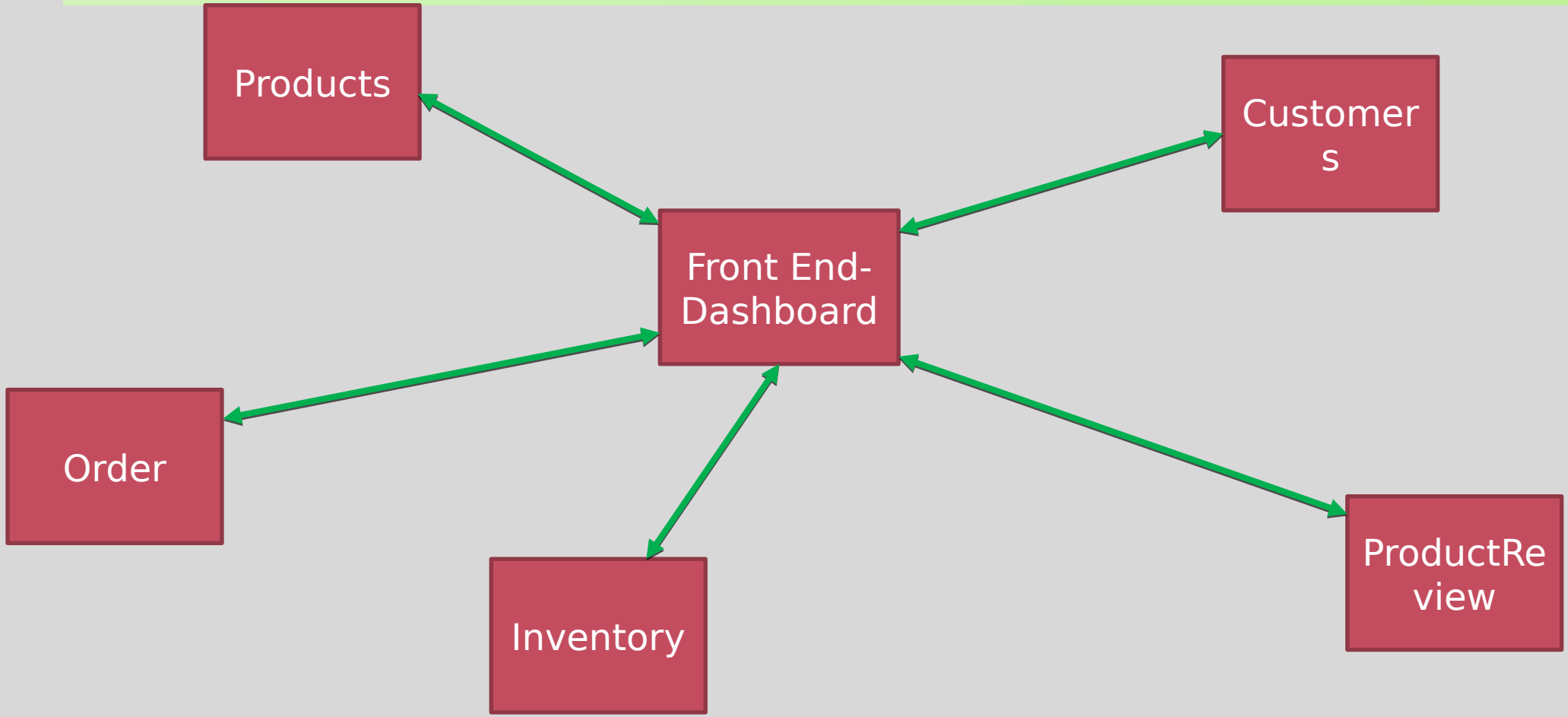
API GATEWAY in microservices

Real life scenario – Shopping App



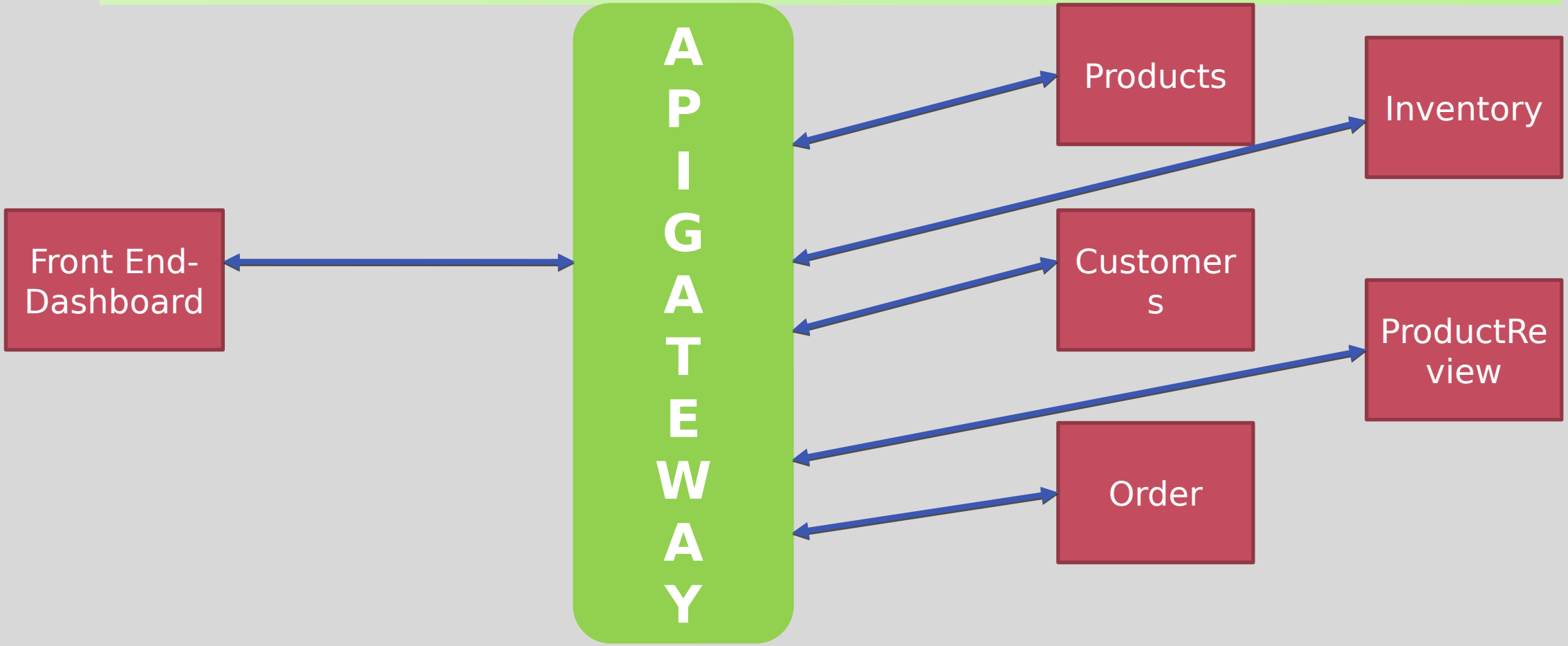
Problem statement

- How the client of microservices can access microservices efficiently and effectively
  - Each client has the address of each microservice?
  - Some mid layer to manage the address of microservices and clients have address of this mid layer service?

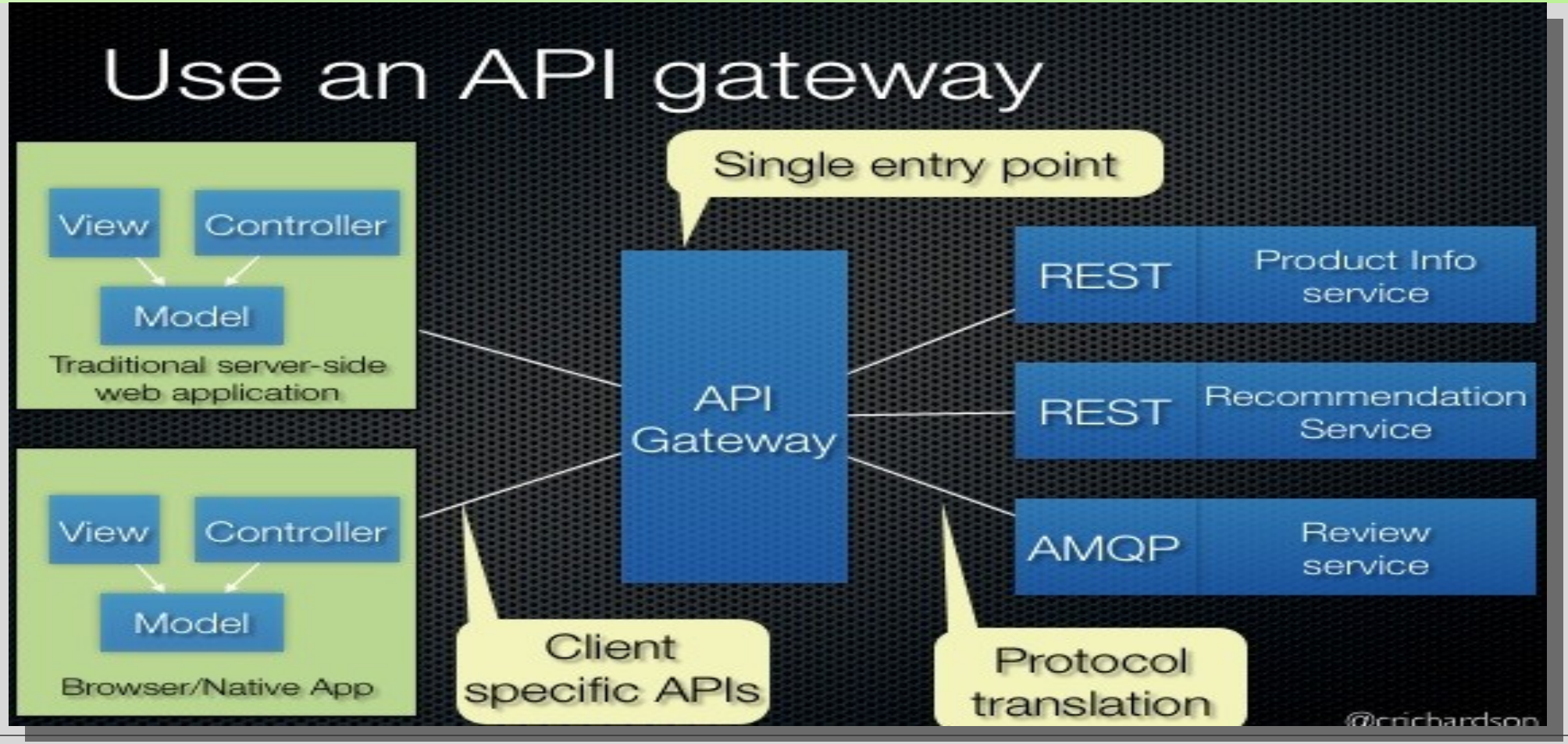
Expectations from Microservices

- Granularity/fine-details of API is deep in microservices.
- Different clients different services
- Latency need of clients
- Network performance
- Adaptability to location and address of microservices change
- Ability to adapt the change is microservices in future
- High availability

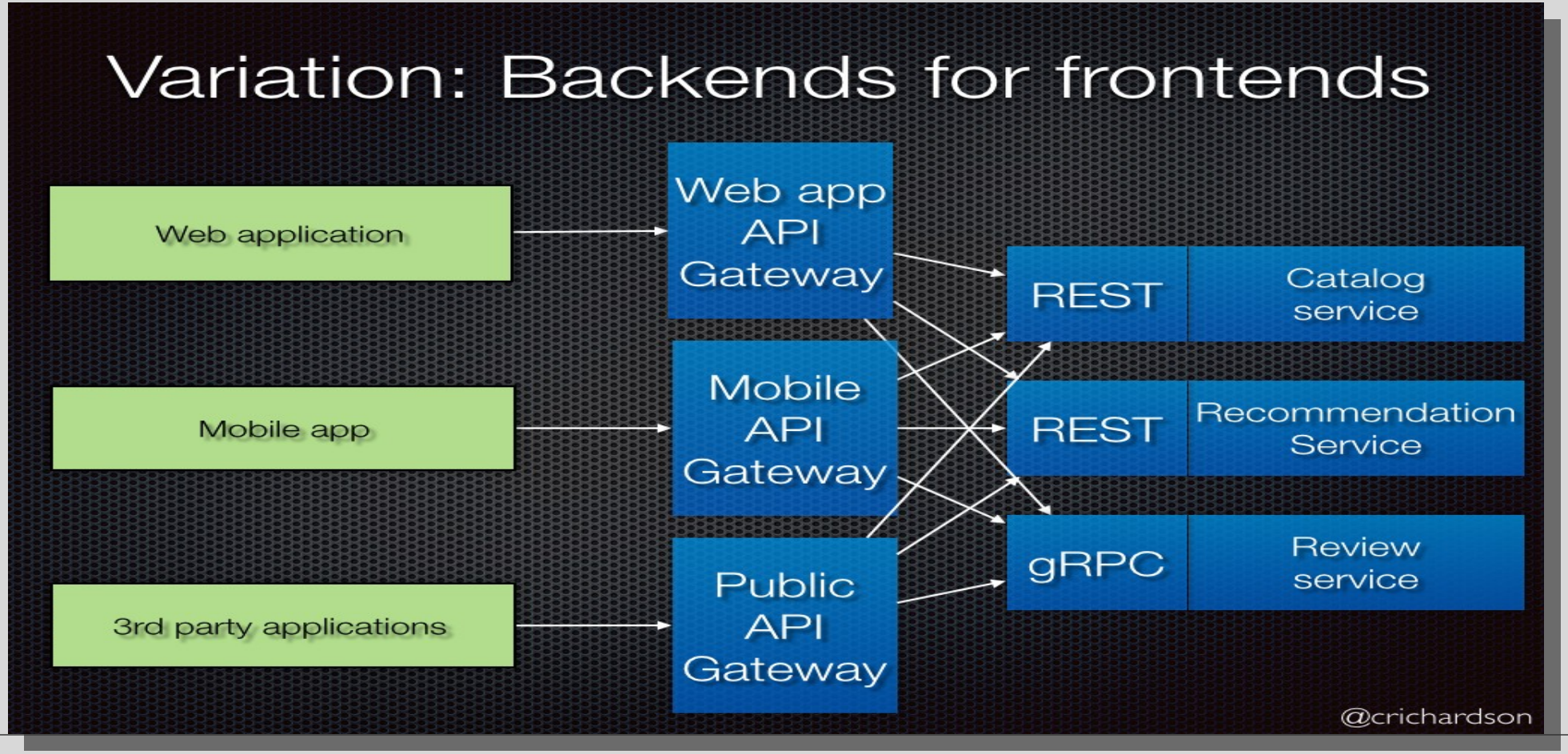
Solution – API gateway



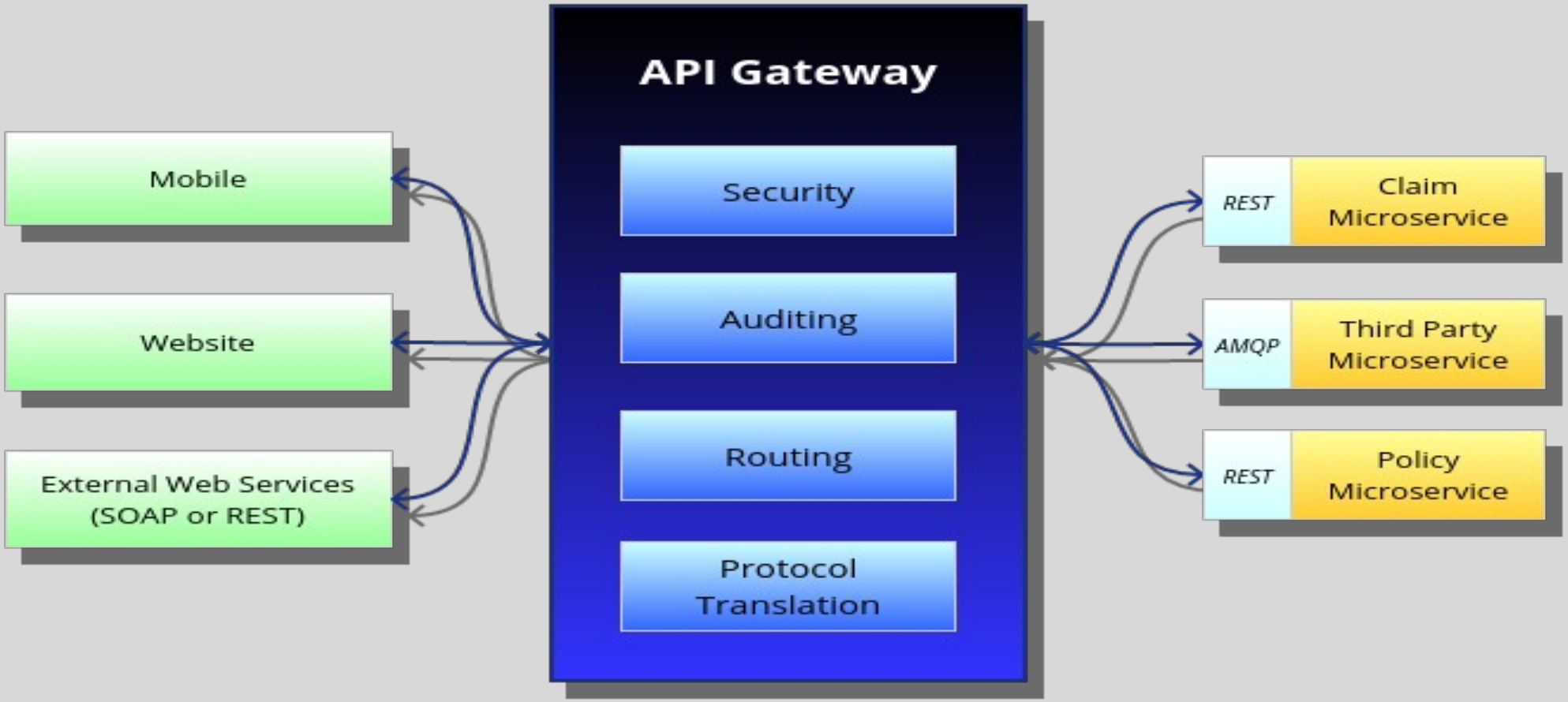
Api gateway..


Use an API gateway
Single entry point
View
Controller
Model
Traditional server-side web application
View
Controller
Model
Browser/Native App
API Gateway
REST
Product Info service
REST
Recommendation Service
AMQP
Review service
Client specific APIs
Protocol translation
@crichardson

Api gateway..


Variation: Backends for frontends
Web application
Mobile app
3rd party applications
Web app API Gateway
Mobile API Gateway
Public API Gateway
REST
Catalog service
REST
Recommendation Service
gRPC
Review service
@crichardson

Api gateway..


API Gateway
Security
Auditing
Routing
Protocol Translation
Mobile
Website
External Web Services (SOAP or REST)
REST
Claim Microservice
AMQP
Third Party Microservice
REST
Policy Microservice

Advantages of API Gateway

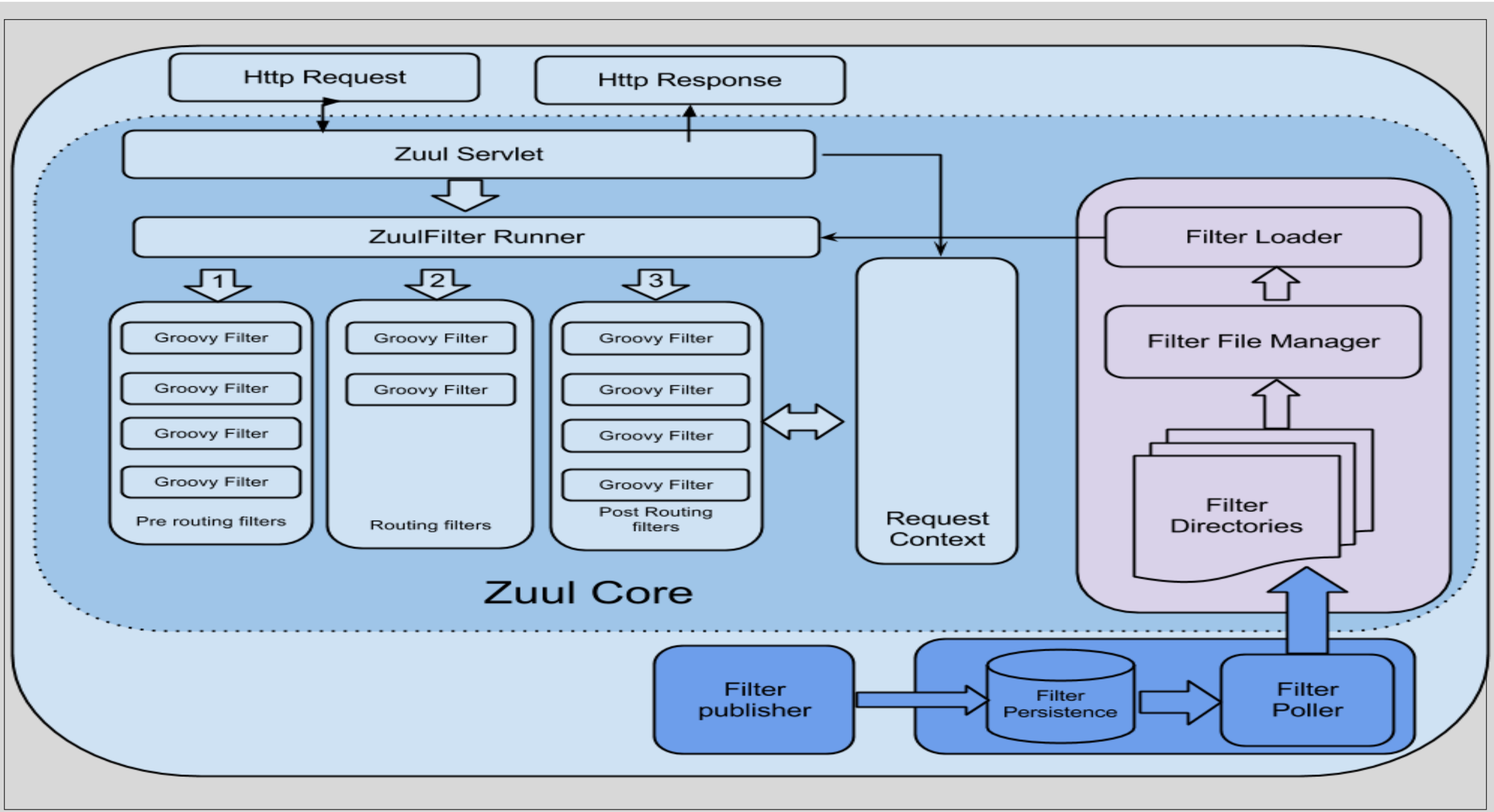
- Separation between clients and microservices
- Simplified clients
- Any change in location of microservices is not going to affect the clients
- Optimal API for each client as per requirement

Drawbacks of API Gateway

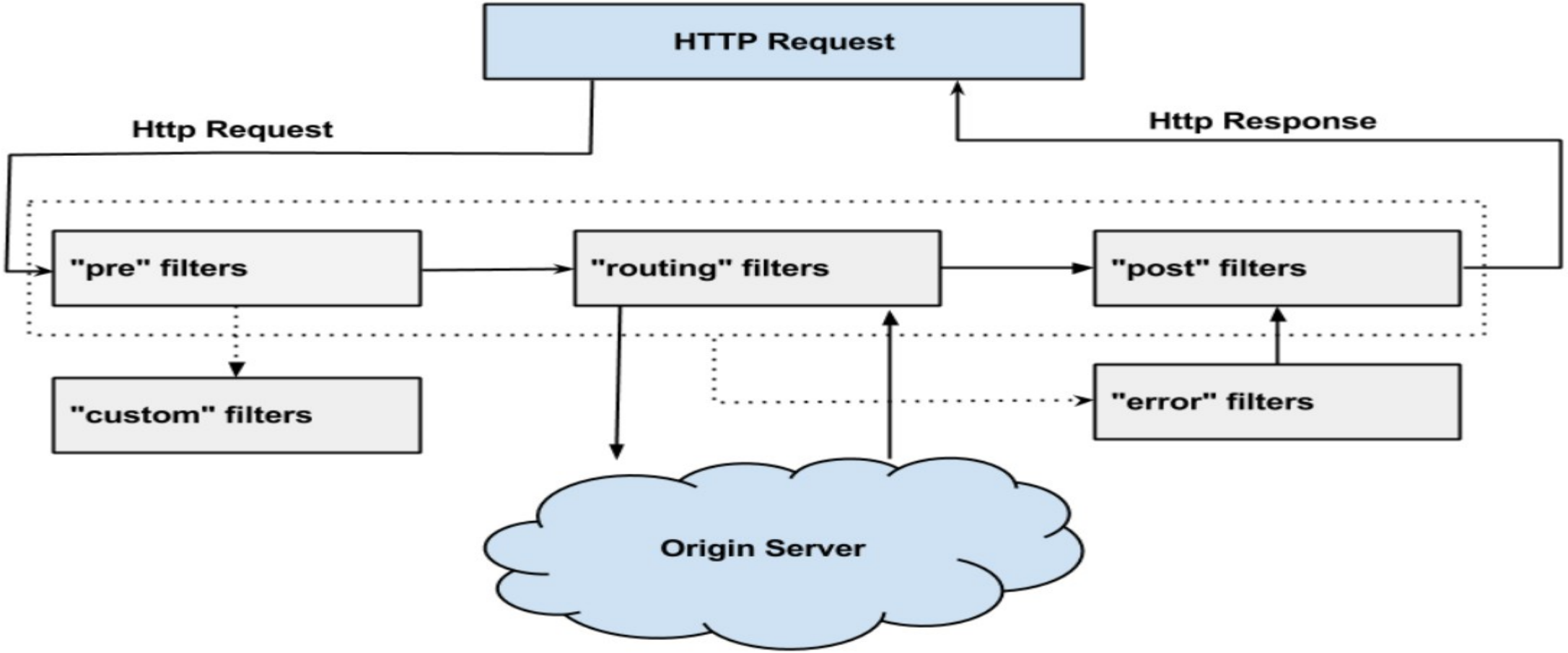
- Complexity
- Latency
- One point failure

API gateway providers for microservices

ZUUL internal architecture


Http Request
Http Response
Zuul Servlet
ZuulFilter Runner
1
2
3
Groovy Filter
Groovy Filter
Groovy Filter
Groovy Filter
Pre routing filters
Groovy Filter
Groovy Filter
Routing filters
Groovy Filter
Groovy Filter
Groovy Filter
Groovy Filter
Post Routing filters
Request Context
Filter Loader
Filter File Manager
Filter Directories
Zuul Core
Filter publisher
Filter Persistence
Filter Poller

Zuul - Filters



Zuul – Filters(pre,post,error,route,custom)

- **Type**
- **Execution Order**
- **Criteria**
- **Action**

Management Endpoints

- https://cloud.spring.io/spring-cloud-netflix/multi/multi__router_and_filter_zuul.html#_management_endpoints

```yaml
management:
  endpoints:
    web:
      exposure:
        include: '*'
  endpoint:
    health:
      show-details: ALWAYS
```

Zuul and Circuit breaker pattern

- https://cloud.spring.io/spring-cloud-netflix/multi/multi__router_and_filter_zuul.html#_management_endpoints

Zuul and spring security

- https://cloud.spring.io/spring-cloud-netflix/multi/multi__router_and_filter_zuul.html#_management_endpoints